Recognition of Visual Dynamical Processes: Theory, Kernels, and Experimental Evaluation

Rizwan Chaudhry and René Vidal

Abstract

Over the past few years, several papers have used Linear Dynamical Systems (LDS)s for modeling, registration, segmentation, and recognition of visual dynamical processes, such as human gaits, dynamic textures and lip articulations. The recognition framework involves identifying the parameters of the LDSs from features extracted from a training set of videos, using metrics on the space of dynamical systems to compare them, and combining these metrics with different classification methods. Usually, each paper makes an ad-hoc choice for every step, and tests the recognition framework on small data sets often involving only one application. We present a detailed evaluation of the LDS-based recognition pipeline; comparing identification methods, metrics, and classification techniques. We propose new metrics that have certain invariance properties and explore a number of variations to the existing metrics. We perform experimental evaluations on well-known data sets of human gaits, dynamic textures, and lip articulations and provide benchmark recognition results. We also analyze the robustness of the recognition pipeline with respect to changes in observation and experimental conditions. Overall, this work represents the most extensive to-date evaluation of the LDS-based recognition framework.

Index Terms

Dynamic Textures, action recognition, linear dynamical systems, kernels for time series data, classification.

I. Introduction

Recognition of nonrigid dynamical scenes – videos of human activities, dynamic textures and lip articulation etc. – has attracted a lot of attention in computer vision over the past years. Traditional object recognition methods deal with recognizing objects and scenes present in individual images. Even though the state-of-the-art of object recognition algorithms like [1] and [2] can deal with some degree of variation in the appearance, pose and scale of the object, they are not explicitly designed to account for variations

in time according to the object's dynamics. For example, in the case of human activity recognition from video, existing object recognition algorithms like [3] and [4] would be able to detect a human in an individual frame and various human poses in a sequence of frames. However, such an algorithm would not account for the temporal dynamics of the pose that characterizes the specific activity.

Linear Dynamical Systems (LDS)s provide a powerful and general framework for modeling the temporal evolution of dynamic scenes. Given a video sequence with some dynamic phenomenon, the first task is to extract features that are representative of the phenomenon e.g., joint angle trajectories for human motion, intensity profiles of video frames in the case of dynamic textures or point trajectories in the case of lip articulation. These features are assumed to be the output of a LDS i.e., features observed at a specific instance of the visual dynamical process are modeled as a linear transformation of the successive states of a linear Gauss-Markov stochastic process. The second task in the recognition pipeline is to identify the system parameters of the LDS that generates these feature trajectories. The third task is to define a distance or *metric* between the system parameters that allows comparison of these processes. Such a metric is immediately useful to perform recognition of novel video sequences by performing, for example, k-nearest neighbors classification.

A large body of literature (see §I-A) exists on the recognition of specific dynamic visual processes, but there has never been a comprehensive evaluation that performs an analysis of the recognition pipeline and provides the best choice of system identification method, metric and the classification method in a generic setting. This paper aims to address this very question.

A. Prior work

Several papers have used LDSs to model dynamic visual phenomena. Some of the significant applications have been synthesis, segmentation and recognition of human gaits, dynamic textures, face motions and lip articulations.

LDSs are generative parametric models. Once the model parameters for a system have been identified, novel sequences can be synthesized. A number of papers have exhibited excellent synthesis of dynamic textures from the learnt parameters [5], [6], [7]. In [8], similar ideas were used for the synthesis of lip articulation with speech as the driving input. For segmenting dynamic textures, a model-based approach was

developed by [9] where an EM-based method was used to segment sequences containing multiple dynamic textures. An alternate variational framework using level sets was proposed in [10] for dynamic texture segmentation. This was further extended in [11] for the moving-boundary case using Ising descriptors.

Towards recognition of visual dynamical phenomena, [12] combined LDSs with Hidden Markov Models (HMMs) to construct a multi-stage algorithm for recognizing human dynamics in video sequences. In [13] an Auto-Regressive Moving Average (ARMA) model was used to classify particular human gaits in videos, such as walking, running, etc. In [14], a method whereby oscillatory gestures, such as travel ahead, travel back, etc. are modeled with dynamical systems was presented. The authors also implemented a predictor module in hardware to recognize these gestures. LDSs were used to classify videos of dynamic textures, such as water, smoke, fluttering flags, etc. in [5]. In [15], ARMA models were used to estimate the appearance of moving faces. Similarly, LDSs have been employed to perform recognition of lip articulation, e.g., in [16] a system theoretic approach was proposed for categorizing lip-articulation from videos of people uttering their names and a six-digit password.

All the aforementioned papers on the recognition of dynamic visual processes use a three-step pipeline: (1) identification of model parameters, (2) computation of a comparison metric between models and (3) classification methods applied to these computed metrics. The method of choice for each of these steps has, for the most part, been ad-hoc and one that is applicable only to each particular application.

The choice of the metric is an important consideration. As the space of LDSs is not Euclidean, defining a good metric is not a trivial task. In most cases, a Riemannian metric is not tractable, as there is no closed form solution for such a metric on the manifold of system parameters. Hence, a number of *cord* metrics have been defined. These include metrics ranging from a geometric notion of subspace angles between the *observability subspaces* spanned by the outputs of the LDSs [17], to a purely algebraic notion of the Binet-Cauchy kernels derived from inner products between outputs of the systems [18], to an information theoretic approach using the KL-divergence between the probability distributions of the output processes [9]. Choosing the best metric and the best classification method for any given application, remains an important and open question that has not been addressed in existing literature. An abundance of metrics also raises the question of whether there are any theoretical connections between seemingly differently derived metrics.

Similarly, for the classification step, [13] and [5] used k-NN to perform recognition of novel sequences of human gaits and dynamic textures, respectively, whereas, kernels based on the metrics above are combined with Support Vector Machine (SVM) in [16], [9]. Recently [19] proposed DynamicBoost – a set of weak dynamic classifiers combined with Adaboost to perform recognition of dynamic textures.

Another frequently unaddressed issue is that of the robustness of the recognition pipeline for visual dynamical processes. In particular, how robust is a certain metric to changes in illumination, pose, scale and frequency etc. of the original visual process? In dynamic texture recogntion, specifically fire recognition, we might not be interested in discriminating between different views of fire but rather only in recognizing fire. Hence invariance to the above mentioned conditions is desirable for this application. On the other hand, in certain human activity recognition applications, we might want a metric that discriminates between different poses of the activity. For example, is the person running sideways, running away from the camera or towards the camera? In this case, we therefore want a metric that is not invariant to changes in pose. This leads to the open question: which metrics are robust to changes in these conditions and which are not? Other than [20], that looked into methods for recognizing non-overlapping dynamic textures that have the same dynamics without considering the appearance of the scene, there has been little work in this direction.

B. Paper contributions

This paper aims to provide a detailed experimental evaluation of the recognition pipeline. We note that even though this paper follows in the footsteps of [13], [21], [15], [9], [18], we consider a number of very important topics that have often been overlooked in the previous approaches. Specifically, our contributions fall into three major categories:

1) We propose three new metrics based on the theory of the Binet-Cauchy kernels [18] and compare their performance to existing metrics. We also explore a number of variations in defining the metrics on LDSs and present methods in which the current metrics can be made more powerful. We show theoretical results that show how kernels based on the subspace angles are in fact special cases of the Binet-Cauchy kernels.
2) We present benchmark results on standard datasets that allow systematic comparison of any newly

developed metrics on the space of LDSs. We provide results for human gait, dynamic texture and lip articulation recognition, which are important problems in computer vision.

3) We propose a set of synthetic experiments that evaluate the performance of recognition algorithms under variations in observation and experimental conditions. Specifically, we analyze the variation in recognition accuracy with changes in the scale, frequency and noise-level of the observations. We also investigate how the recognition accuracy is affected by the length of individual sequences used for learning the system parameters, the amount of training data and the number of classes.

C. Paper outline

We briefly review the LDS model in §II and describe how the parameters of a system that generates the given output feature trajectories can be identified. In §III, we define a number of metrics that have been used for comparing LDSs. In §IV, we describe two methods for classification of LDSs based on these metrics. We delve into the important theoretical considerations as well as propose new kernels and powerful improvements to the current ones in §V. Finally, in §VI and §VII, we present the experimental evaluation of the recognition pipeline and give concluding remarks in §VIII.

II. Linear dynamical systems

As mentioned earlier, nonrigid scenes are modeled by treating their representative feature trajectories, $\{\mathbf{y}_t\}_{t=1}^N$, as the output of a LDS. In the case of dynamic textures, these can be the image intensities, while in the case of human gaits, these can be 3-D trajectories of motion capture data or variations in joint angles through time. In the case of lip articulation these can be trajectories of landmarks extracted from videos of lip motion. In particular, one assumes these features to be the output of a Linear Dynamical System (LDS) that is represented by the model $\mathbf{M} = (\mathbf{x}_0, \mu, A, B, C, R)$ and the equations,

$$\begin{aligned} \mathbf{x}_{t+1} &= A\mathbf{x}_t + B\mathbf{v}_t \\ \mathbf{y}_t &= \mu + C\mathbf{x}_t + \mathbf{w}_t. \end{aligned} \tag{1}$$

In eq. (1), $\mathbf{x}_t \in \mathbb{R}^n$ is the state of the LDS at time t and represents the internal dynamics of the system; $\mathbf{y}_t \in \mathbb{R}^p$ is the observed output trajectory at time t; $\mathbf{x}_0$ is the initial state of the system; and $\mu \in \mathbb{R}^p$ is the mean of $\{\mathbf{y}_t\}_{t=1}^N$, e.g., the mean frame of video, the mean joint angle configuration

etc. $A \in \mathbb{R}^{n\times n}$ describes the dynamics of the state evolution, $B \in \mathbb{R}^{n\times n_v}$ models the way in which input noise affects the state evolution and $C \in \mathbb{R}^{p\times n}$ transforms the state to an output or observation of the overall system. $\mathbf{v}_t \in \mathbb{R}^{n_v}$ and $\mathbf{w}_t \in \mathbb{R}^p$ are the system noise and the observation noise at time t, respectively. We assume that the noise processes are zero-mean i.i.d. Gaussian, such that $\mathbf{v}_t \sim G(\mathbf{v}_t, 0, I_{n_v})$ and $\mathbf{w}_t \sim G(\mathbf{w}_t, 0, R), R \in \mathbb{R}^{p\times p}$. By this definition, $B\mathbf{v}_t \sim G(B\mathbf{v}_t, 0, Q)$ where $Q = BB^\top \in \mathbb{R}^{n_v\times n_v}$ and $G(\mathbf{z}, \mu_\mathbf{z}, \Sigma) = (2\pi)^{-n/2}|\Sigma|^{-1/2}\exp(-\frac{1}{2}\|\mathbf{z}-\mu_\mathbf{z}\|^2_\Sigma)$ is a multivariate Gaussian distribution on $\mathbf{z} \in \mathbb{R}^n$ with $\|\mathbf{z}\|^2_\Sigma = \mathbf{z}^\top\Sigma^{-1}\mathbf{z}$. We also assume that $\mathbf{v}_t$ and $\mathbf{w}_t$ are independent processes. Typically $n_v \leq n$ and in the case of dynamic textures, $p \gg n$.

Given the sequence $\{\mathbf{y}_t\}_{t=0}^{N-1}$, assumed to be the output of a LDS, we need to learn the system parameters, $(\mathbf{x}_0, \mu, A, B, C, R)$, whose outputs best approximate the observed output. We will briefly describe some of the more commonly used identification methods below:

A. Expectation-Maximization (EM) based Identification

This system identification method exploits the fact that LDSs can be interpreted as generative probabilistic models with a hidden state $\mathbf{x}_t$. Therefore, the parameters of the LDS can be identified in a maximum likelihood sense using the EM algorithm. EM iterates between estimating the missing information, i.e., the states $\mathbf{x}_t$, given the current parameters, and computing the new parameters given the latest estimates of the states. The conditional probability of the state distribution and the conditional probability of the output distribution are given by:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = G(\mathbf{x}_t, A\mathbf{x}_{t-1}, Q), \quad \text{and} \quad p(\mathbf{y}_t|\mathbf{x}_t) = G(\mathbf{y}_t, C\mathbf{x}_t, R). \tag{2}$$

If $p(\mathbf{x}_0)$ is the probability of observing the initial state in eq. (1), then the joint distribution of the state and output sequence is given by:

$$p(\mathbf{x}_0^{N-1}, \mathbf{y}_0^{N-1}) = p(\mathbf{x}_0)\prod_{t=1}^{N-1} p(\mathbf{x}_t|\mathbf{x}_{t-1})\prod_{t=0}^{N-1} p(\mathbf{y}_t|\mathbf{x}_t). \tag{3}$$

We apply EM to estimate both the state and the parameters from the output measurements. The estimation step uses Kalman smoothing filters for the calculation of the parameters $\mu_\mathbf{z}$ and Σ for each of the processes in eq. (2). For more details we refer the reader to [22], [9] and the references therein.

B. Numerical algorithms for Subspace State Space System Identification (N4SID)

In [23] and [24], a method is presented for finding *asymptotically* optimal estimates of the system parameters in a maximum likelihood sense, under certain assumptions [25] on the noise processes $\mathbf{v}_t$ and $\mathbf{w}_t$. The main advantage of N4SID over EM is that it provides a closed-form, rather than iterative, solution of the identification problem based on linear-algebraic techniques, using subspace algorithms on the Hankel matrices of the outputs. Its main drawback, however, is that it is not computationally efficient when the dimension of the output data becomes very large. Hence, using N4SID is prohibitive when performing system identification for dynamic textures, where the dimension of the system output is equal to the number of pixels in an image. For human gaits or lip articulations, on the other hand, the dimension of the output data is often small, hence N4SID can be useful for system identification.

C. Principal Component Analysis (PCA) based Identificaiton

Motivated by the fact that N4SID becomes computationally prohibitive when the output dimension is large, there was a need for more computationally efficient methods for identifying the system parameters of dynamic textures. Doretto et al. [5] proposed a method based on PCA that gives a suboptimal, but very fast solution for finding the system parameters of a LDS. In particular, given the output sequence, $\{\mathbf{y}_t\}_{t=0}^{N-1}$, a compact, rank n, singular value decomposition, of the matrix $Y_0^{N-1} = [\mathbf{y}_0 - \mu, \ldots, \mathbf{y}_{N-1} - \mu] = U\Sigma V^\top$ is performed, where $\mu = \frac{1}{N}\sum_{t=0}^{N-1} \mathbf{y}_t$. The system parameters and the state parameters are then estimated as $C = U, \quad X_0^{N-1} = \Sigma V^\top$, where $X_0^{N-1} = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N-1}]$ are the estimated states of the system. Note that in computing X_0^{N-1}, the state equation (1) is not enforced. This is what makes this method sub-optimal. Given the state sequence, the matrix A can be computed using least-squares as $A = X_1^{N-1}(X_0^{N-2})^\dagger$ where $X^\dagger$ represents the pseudo-inverse of X. Also, $Q = \frac{1}{N-1}\sum_{t=0}^{N-2} \mathbf{v}'_t(\mathbf{v}'_t)^\top$ where $\mathbf{v}'_t = B\mathbf{v}_t = \mathbf{x}_{t+1} - A\mathbf{x}_t$.

III. Metrics on the space of LDSs

Given the parameters of LDSs identified from videos of multiple dynamic scenes, we need a method to compare these parameters in order to recognize novel video sequences. Hence, we need to define a metric that provides a measure of (dis)similarity on the space of LDSs. This space is not Euclidean, hence a careful choice of a metric is needed in order to perform a valid and meaningful comparison.

In particular, it is noteworthy that the representation $\mathbf{M} = (\mathbf{x}_0, \mu, A, B, C, R)$ is unique only up to an invertible linear transformation, T, of the state space. Specifically, the model $\mathbf{M}$ is equivalent to $\mathbf{M}_T = (T\mathbf{x}_0, \mu, TAT^{-1}, TB, CT^{-1}, R)$ for all invertible linear transformations $T \in \mathbb{R}^{n\times n}$. Hence the required metrics also need to be invariant with respect to such transformations.

Prior work on comparing LDSs can broadly be classified into three types. (1) metrics based on a purely geometric notion of subspace angles between the *observability subspaces* of the systems [17], (2) algebraic metrics based on the Binet-Cauchy kernels [18], [26] and (3) information theoretic metrics based on the KL-divergence between the probability distributions of the two stochastic processes [9]. The dynamics and output observations of the system are determined by the matrix parameters (A, C) in eq. (1). Hence, for the most part, we concentrate on these parameters and not on (μ, B, R) which determine the linear trends in the data and the effect of the input noise to the state and output. However there has been recent work by Bissacco et al. in [26] that develops metrics specifically taking into consideration the effect of the input and output noise processes. They present a complete recognition pipeline including an ID method as well as a metric for non-minimum phase linear non-Gaussian (Hammerstein) models. Their work deals with a very specific kind of LDS that is not relevant in all general situations, hence we do not consider it in our evaluation. We briefly describe these metrics and then provide a number of variations that we will investigate in our experiments.

A. Distances based on subspace angles

In the absence of noise, the output trajectory, $\mathbf{y}_t$ of a LDS lives in the *observability* subspace associated with the model parameters, $\mathbf{M} = (A, C)$. The observability subspace is the range-space of the extended observability matrix [27], $\mathcal{O}_\infty(\mathbf{M}) = [C^\top, (CA)^\top, (CA^2)^\top, \ldots]^\top \in \mathbb{R}^{\infty\times n}$. Therefore, one way to compare two LDSs is to find the principal angles [28] between these matrices. To find these subspace angles, we first convert the model in eq. (1) to the *forward innovation form* [17] by applying a change of coordinates, $\mathbf{x}'_t = T\mathbf{x}_t$ to the original representation in eq. (1):

$$\begin{aligned} \mathbf{x}'_{t+1} &= A\mathbf{x}'_t + K\mathbf{e}_t \\ \mathbf{y}_t &= C\mathbf{x}'_t + \mathbf{e}_t, \end{aligned} \tag{4}$$

where the Kalman gain, $K = (APC^\top)(CPC^\top + I)^{-1}$, and P is the solution of the Riccati equation:

$$P = APA^\top - (APC^\top)(CPC^\top + I)^{-1}(APC^\top)^\top + Q. \tag{5}$$

Given the equivalent representations, $M_1' = (A_1, K_1, C_1)$ and $M_2' = (A_2, K_2, C_2)$, the subspace angles between M_1 and M_2 are defined as the angles between the range spaces of $[\mathcal{O}_\infty(M_1)\mathcal{O}_\infty(M_2^{-1})]$ and $[\mathcal{O}_\infty(M_2)\mathcal{O}_\infty(M_1^{-1})]$ where $M_i^{-1} = (A_i - K_iC_i, K_i, C_i)$ is the inverse system of M_i. For more details on the computation of these angles, refer to [17]. Once the subspace angles, $\{\theta_i\}_{i=1}^{2n}$, have been determined, a number of metrics can be defined between $\mathbf{M}_1$ and $\mathbf{M}_2$:

$$\text{Finsler distance:} \quad d_F(\mathbf{M}_1, \mathbf{M}_2) = \theta_{\max}, \tag{6}$$

$$\text{Gap distance:} \quad d_g(\mathbf{M}_1, \mathbf{M}_2) = \sin\theta_{\max}, \tag{7}$$

$$\text{Frobenius distance:} \quad d_f(\mathbf{M}_1, \mathbf{M}_2)^2 = 2\sum_{i=1}^{2n} \sin^2\theta_i, \tag{8}$$

$$\text{Martin distance:} \quad d_M(\mathbf{M}_1, \mathbf{M}_2)^2 = -\ln\prod_{i=1}^{2n} \cos^2\theta_i, \tag{9}$$

$$\text{Martin kernel:} \quad k_M(\mathbf{M}_1, \mathbf{M}_2) = \prod_{i=1}^{2n} \cos^2\theta_i. \tag{10}$$

In [29], it was shown that the Martin kernel in eq. (10) is a positive definite kernel and thus computes an inner product in a feature space of the original LDSs. Notice that the Gap distance is a monotonically increasing function of the Finsler distance on the domain $[0, \pi]$ and hence these distances are the same from a classification perspective.

B. Distances based on Binet-Cauchy kernels

Vishwanathan et al. [18] introduced an algebraic approach to comparing two LDSs, leadING to a complete family of kernels called the *Binet Cauchy* kernels. One of the proposed kernels is the *trace* kernel which is derived from an infinite series of inner products on the output sequences of two systems. Specifically, $k_{tr}(\mathbf{M}_1, \mathbf{M}_2) = \sum_{t=0}^{\infty} \lambda^t \mathbf{y}_{t;1}^\top W \mathbf{y}_{t;2}$ where W is a positive definite weight matrix and $|\lambda| < 1$ is a discounting factor. It was shown in [18] that when the initial conditions are deterministic, the *trace* kernel for LDSs can be written as

$$k_{tr}(\mathbf{M}_1, \mathbf{M}_2) = \mathbf{x}_{0;1}^\top P \mathbf{x}_{0;2} + \frac{\lambda}{1-\lambda}\text{trace}(B_1^\top P B_2), \tag{11}$$

under the assumptions that $\mathbb{E}[\mathbf{x}_{0;1}] = \mathbb{E}[\mathbf{x}_{0;2}] = \mathbf{0}, 0 < \lambda < 1$ where $\mathbf{x}_{t;i}$ represents the state at time t of the system $\mathbf{M}_i$. . The matrix $P \in \mathbb{R}^{n \times n}$ is the solution to the Sylvester equation $P = \lambda A_1^\top P A_2 + C_1^\top C_2$. However, when the initial conditions are not deterministic, $\mathbf{x}_{0;1}$ and $\mathbf{x}_{0;2}$ are random variables with covariance $\Sigma_{\mathbf{x}_0} = \mathbb{E}[\mathbf{x}_{0;1}\mathbf{x}_{0;2}^\top]$, the trace kernel can now be written as,

$$k_{tr}(\mathbf{M}_1, \mathbf{M}_2) = \text{trace}(\Sigma_{x_0} P) + \frac{\lambda}{1-\lambda}\text{trace}(B_1^\top P B_2). \tag{12}$$

The trace kernel is normalized so that $k_{tr}(\mathbf{M}_i, \mathbf{M}_i) = 1$ by computing

$$k_T(\mathbf{M}_1, \mathbf{M}_2) = \frac{k_{tr}(\mathbf{M}_1, \mathbf{M}_2)}{\sqrt{k_{tr}(\mathbf{M}_1, \mathbf{M}_1)}\sqrt{k_{tr}(\mathbf{M}_2, \mathbf{M}_2)}}. \tag{13}$$

One can also create a distance from the trace kernel by defining

$$d_{k_T}(\mathbf{M}_1, \mathbf{M}_2)^2 = k_T(\mathbf{M}_1, \mathbf{M}_1) + k_T(\mathbf{M}_2, \mathbf{M}_2) - 2k_T(\mathbf{M}_1, \mathbf{M}_2), \tag{14}$$

In [18], the Binet-Cauchy *determinant* kernels for LDSs were introduced as:

$$k_{\text{det}}(\mathbf{M}_1, \mathbf{M}_2) := \mathbb{E}_{\mathbf{v},\mathbf{w}} \det \left[\sum_{t=0}^{\infty} \lambda^t \mathbf{y}_{t;1} (\mathbf{y}'_{t;2})^\top \right]. \tag{15}$$

However these determinant kernels become very computationally unweildy with high dimensional output data and hence are of very limited use in practical problems. We will not consider these determinant kernels in our evaluation.

C. Distances based on the KL-divergence

Chan et al. in [9], introduced an information theoretic approach for defining a metric between two LDSs. Since the output of a LDS can be interpreted as a stochastic process, the well known Kullback-Leibler (KL) divergence, $D(P_{Y_1} \| P_{Y_2})$ between the probability distributions of the mean subtracted output processes $Y_1 = [\mathbf{y}_{i;1} - \mu_1 \in \mathbb{R}^m]_{i=0}^{\infty}$ and $Y_2 = [\mathbf{y}_{i;2} - \mu_2 \in \mathbb{R}^m]_{i=0}^{\infty}$ provides a (dis)similarity metric between two LDSs. However, since the sequences Y_1 and Y_2 have infinite length, only an approximation to the true KL divergence can be computed from the available data, $Y_1^{0:N-1} = [\mathbf{y}_{0;1} - \mu_1, \ldots, \mathbf{y}_{N-1;1} - \mu_1]$ and $Y_2^{0:N-1} = [\mathbf{y}_{0;2} - \mu_2, \ldots, \mathbf{y}_{N-1;2} - \mu_2]$ as:

$$D(P_{Y_1} \| P_{Y_2}) \approx \frac{1}{N} D(P_{Y_1^{0:N-1}} \| P_{Y_2^{0:N-1}}). \tag{16}$$

For a LDS, the output sequence is distributed as $Y_i^{0:N-1} \sim G(Y_i, \Upsilon_i, \Phi_i), i = 1, 2$. Omitting i, $\Upsilon = \underline{C}[(\mathbf{x}_0^\top, (A\mathbf{x}_0)^\top, \ldots, (A^{N-1}\mathbf{x}_0)^\top]^\top$ is the mean of $Y^{0:N-1}$ and $\Phi = \underline{C}\Sigma\underline{C}^\top + \underline{R}$ is the covariance matrix.

Here $\underline{C}$ and $\underline{R}$ are block diagonal matrices of appropriate dimensions formed from C and R, respectively, and Σ is a matrix with block entries $\Sigma_{(i,j)} = A^{i-j}\sum_{k=0}^{\min(i,j)} A^k Q (A^k)^\top$ for $j \leq i \leq N$, and $\Sigma_{(j,i)} = \Sigma_{(i,j)}^\top$ for $i < j$. The KL divergence between two output sequences can be computed as

$$D(P_{Y_1^{0:N-1}} \| P_{Y_2^{0:N-1}}) = \frac{1}{2}[\log \frac{\det \Phi_2}{\det \Phi_1} + \text{trace}(\Phi_2^{-1}\Phi_1) + \|\Upsilon_1 - \Upsilon_2\|_{\Phi_2}^2 - pN], \tag{17}$$

where p is the output dimension. The KL divergence is made symmetric to obtain the KL divergence distance as shown below. For more details, refer to [9].

$$d_{KL}(\mathbf{M}_1, \mathbf{M}_2) = \frac{1}{2}[D(P_{Y_1^{1:N}} \| P_{Y_2^{1:N}}) + D(P_{Y_2^{1:N}} \| P_{Y_1^{1:N}})]. \tag{18}$$

IV. Classification methods

The metrics in §III provide a method of performing a comparison between two LDSs, $\mathbf{M}_i$ and $\mathbf{M}_j$. Given a set of labeled video sequences with dynamic phenomena, one can learn the parameters, $\mathbf{M}_1, \ldots, \mathbf{M}_N$, of the LDSs that generates each sequence and hence compute all pair-wise (dis)similarities. Given a novel sequence, the final task is to recognize it as belonging to one of the already learnt classes. We will briefly discuss two of the most common classification algorithms used in this context.

A. k-nearest neighbors

Given a labeled training set of LDSs, the k-nearest neighbor classification method finds the distances of a novel model from all the models in the training set. The k models in the training set that are nearest to this novel model is then chosen. The class shared by the majority of these models is assigned to the novel model. More specifically, given the list of system-class pairs in the training data, $T = \{(\mathbf{M}_1, g_1), (\mathbf{M}_2, g_2), \ldots, (\mathbf{M}_N, g_N)\}$, where $g_i \in \{1, \ldots, m\}$ for an m-class problem, the class, g_t, for a novel test system $\mathbf{M}_t$ is determined by finding all the distances $d(\mathbf{M}_i, \mathbf{M}_t),\ i = 1, \ldots, N$ and finding the k nearest systems. The most commonly used value of $k = 1$ yields $s = \underset{j=1,\ldots,N}{\operatorname{argmin}}\, d(\mathbf{M}_j, \mathbf{M}_t)$ as the index of the closest system in the training set. Hence, the class for $\mathbf{M}_t$ is given by $g_t = g_s$.

B. Support vector machine (SVM)

Given training data, $\{(\mathbf{s}_1, g_1), \ldots, (\mathbf{s}_N, g_N)\}$, where $\mathbf{s}_i \in \mathcal{S}$ are data points in a Euclidean space, $\mathcal{S}$, and $g_i \in \{-1, +1\}$ are binary labels, the objective of SVM is to find a hyperplane, $\{\mathbf{s} \in \mathcal{S} | w^\top \mathbf{s} + b = 0\}$,

that separates the two classes by maximizing the distance between the data points and the separating hyperplane. If the data are linearly separable, an optimal hyperplane can be determined as $w = \sum_{i=1}^{N} \beta_i \mathbf{s}_i$, where $\beta_i \neq 0$ for only a few data points called the *support vectors*. For a new test data point $\mathbf{s}_t$, the binary class membership is determined by computing $g_t = \text{sign}(\sum_{i=1}^{N} \beta_i \mathbf{s}_i^\top \mathbf{s}_t + b)$. The binary classifier, however, can be extended to the multi-class case using a one-against-all approach [30].

If the data are not linearly separable, then slack variables are used to penalize inaccurate class memberships. The SVM algorithm then maximizes the margin between the classes by keeping the slack to a minimum. Kernel SVM, on the other hand, employs the approach of using an embedding function, $\Phi : \mathcal{S} \rightarrow \mathcal{F}$ to embed the non-separable data into a higher dimensional feature space, $\mathcal{F}$, where the data is assumed to be linearly separable. Any new test point can thus be classified by computing $g_t = \text{sign}(\sum_{i=1}^{N} \beta_i k(\mathbf{s}_i, \mathbf{s}_t) + b)$ where $k(\mathbf{s}_i, \mathbf{s}_j) = \Phi(\mathbf{s}_i)^\top \Phi(\mathbf{s}_j)$ is the *kernel* for the SVM computation.

For LDSs, the data resides in a manifold. Therefore, in order to use SVM for classifying LDSs, we need kernels in the space of LDSs. Any of the kernels described in §III can be used for this purpose. A test system $\mathbf{M}_t$ is classified by computing $g_s = \text{sign}(\sum_{i=1}^{N} \hat{\beta}_i k(\mathbf{M}_i, \mathbf{M}_t) + \hat{b})$ where $\hat{\beta}$ and $\hat{b}$ are learnt during the training phase of the SVM algorithm.

V. Analysis of existing metrics and extensions

Having general metrics with tunable parameters gives the system-designer much more power in terms of giving more weight to certain properties of the system and less to others when making comparisons, thereby making metrics that are more specific to the application at hand. For example, free parameters in the expression for the metric can be tuned to the training data using cross-validation during the training stage of SVMs.

In this section, we will propose numerous improvements to the existing metrics discussed in §III. We will first propose several ways of making the Binet Cauchy kernels invariant to the initial conditions of the systems. We will also derive theoretical connections between the Binet Cauchy kernels and the kernels based on subspace angles. Finally, we will propose how the effects of the initial conditions, the input noise, the observation mean and linear trends in the data can be incorporated in the computation of the metric.

A. The effect of initial conditions

For some particular applications, it might be desirable to have a classifier that depends on the initial state, while for some others this might not be the case. Out of the metrics presented in §III, notice that the KL divergence and the Binet Cauchy kernels depend on the initial conditions, while the subspace angles based metrics do not depend on the initial conditions. Hence it is desirable to also make the trace kernel independent of the initial condition as well as independent of the noise process. For example, only the dynamics and appearance of the texture is generally of interest when analyzing dynamic textures. Moreover, as outlined in [26], in the case of periodic motions, the initial condition affects the phase of the trajectory and the resulting metric is not consistent across different initial conditions of the same motion class. One way proposed in [18] by which the Binet Cauchy kernels can be made invariant to initial conditions is by using the covariance matrix of the distribution of the initial conditions as stated in eq. (12). However, this information is not always available or deducible from the data and hence there is a need to develop methods that do not require any statistics of the initial conditions.

In this section, we propose the following three ways in which invariance to initial conditions can be obtained for the first term in the Binet Cauchy kernels:

$$\text{Initial state-independent trace kernel:} \qquad k_t(\mathbf{M}_1, \mathbf{M}_2) = \text{trace}(P), \tag{19}$$

$$\text{Determinant kernel}^1\text{:} \qquad k_d(\mathbf{M}_1, \mathbf{M}_2) = |\det(P)|, \tag{20}$$

$$\text{Maximum singular value kernel:} \qquad k_\sigma(\mathbf{M}_1, \mathbf{M}_2) = \sigma_{\max}(P), \tag{21}$$

where P is the solution of the Sylvester equation $P = \lambda A_1^\top P A_2 + C_1^\top C_2$. It is however important to note the inital state-independent trace kernel, k_t, and the maximum singular value kernel, k_σ, are not invariant to arbitrary basis transformations, T_1 and T_2, of the original system parameters $\mathbf{M}_1$ and $\mathbf{M}_2$, respectively, and hence do not constitute valid metrics as such. This can easily be seen from the following analysis for the initial state-independent trace kernel:

$$\begin{aligned} P' &= \lambda (T_1 A_1 T_1^{-1})^\top P' (T_2 A_2 T_2^{-1}) + (C_1 T_1^{-1})^\top (C_2 T_2^{-1}), \\ \text{trace}(P') &= \lambda \text{trace}(T_1^{-T} A_1^\top T_1^\top P' T_2 A_2 T_2^{-1}) + \text{trace}(T_1^{-\top} C_1^\top C_2 T_2^{-1}) \neq \text{trace}(P), \end{aligned} \tag{22}$$

[1]Hereforth whenever we mention the determinant kernel, we refer to this initial-state-independent definition. The original Binet-Cauchy determinant kernel in [18] briefly mentioned in §IV-B is computationally unwieldy and is not used in this paper.

i.e., a change in basis transformation for any of the systems will give a different value for the trace kernel. One way to overcome this limitation is to convert all the system parameters to some canonical form like those proposed in [31] and then computing the kernels on these canonical parameters. We use the *Jordan Canonical Form* proposed in [31] for our experiments since it is numerically stable. The normalized determinant kernel, however, is independent of basis change transformations as we show in the following theorem.

Theorem 5.1: The normalized Binet-Cauchy determinant kernel, $k'_d(\mathbf{M}_1, \mathbf{M}_2)$ is independent to basis transformations T_1 and T_2 of $\mathbf{M}_1$ and $\mathbf{M}_2$ respectively.

Proof: From eq. (20) we can see that $k_d(\mathbf{M}_i, \mathbf{M}_j) = |\det(P_{ij})|$, where $P_{ij} = \lambda A_i^\top P_{ij} A_j + C_i^\top C_j$. Under basis transformations T_1 and T_2 of $\mathbf{M}_1$ and $\mathbf{M}_2$ respectively,

$$P'_{ij} = \lambda (T_i A_i T_i^{-1})^\top P'_{ij} (T_j A_j T_j^{-1}) + (C_i T_i^{-1})^\top (C_j T_j^{-1}),$$
$$\Rightarrow T_i^\top P'_{ij} T_j = \lambda A_i^\top (T_i^\top P'_{ij} T_j) A_j + C_i^\top C_j = P_{ij}.$$

The normalized determinant kernel,

$$\begin{aligned} k'_d(\mathbf{M}'_1, \mathbf{M}'_2) &= \frac{|\det P'_{12}|}{\sqrt{|\det P'_{11}|}\sqrt{|\det P'_{22}|}} \\ &= \frac{|\det(T_1^{-\top} P_{12} T_2^{-1})|}{\sqrt{|\det(T_1^{-\top} P_{11} T_1^{-1})|}\sqrt{|\det(T_2^{-\top} P_{22} T_2^{-1})|}} \\ &= \frac{|\det P_{12}|}{\sqrt{|\det P_{11}|}\sqrt{|\det P_{22}|}} = k'_d(\mathbf{M}_1, \mathbf{M}_2). \end{aligned} \tag{23}$$

Hence the normalized determinant kernel,

$$\text{Normalized determinant kernel:} \quad k'_d = \frac{k_d(\mathbf{M}_1, \mathbf{M}_2)}{\sqrt{k_d(\mathbf{M}_1, \mathbf{M}_1)}\sqrt{k_d(\mathbf{M}_2, \mathbf{M}_2)}}, \tag{24}$$

is independent of basis transformations of the system parameters and hence a valid kernel for comparison between LDSs that is invariant to the initial conditions. ■

B. Connections between different metrics

We now delve into the relationship between the Binet-Cauchy kernels and the subspace-angles based kernels.

Theorem 5.2: The normalized Binet-Cauchy determinant kernel, k'_d, for $\lambda = 1$, coincides with the Martin kernel, k_M, for Auto Regressive (AR) models.

Proof: For the determinant kernel, $k_d(\mathbf{M}_1, \mathbf{M}_2) = |\det(P)|$, note that if we set $\lambda = 1$, then P becomes the solution of the Lyapunov equation, $P = A_1^\top P A_2 + C_1^\top C_2$. Now, following [32], [17], the calculation of the subspace angles between two AR models is performed by first solving the Lyapunov equation, $\mathcal{A}^\top \mathcal{Q} \mathcal{A} - \mathcal{Q} = -\mathcal{C}^\top \mathcal{C}$ for, $\mathcal{Q} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$, where $\mathcal{A} = \begin{bmatrix} A_1 & \mathbf{0} \\ \mathbf{0} & A_2 \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$ and $\mathcal{C} = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \in \mathbb{R}^{p \times 2n}$. The n largest eigenvalues $\{\lambda_i\}_{i=1}^n$ of $\begin{bmatrix} \mathbf{0} & P_{11}^{-1} P_{12} \\ P_{22}^{-1} P_{21} & \mathbf{0} \end{bmatrix}$ are equal to the cosines of the subspace angles $\{\theta_i\}_{i=1}^n$, i.e.,

$$\cos^2 \theta_i = i\text{-th eigenvalue}(P_{11}^{-1} P_{12} P_{22}^{-1} P_{21}). \tag{25}$$

Notice that P, the solution of the Lyapunov equation for the Binet Cauchy kernels, is equal to P_{12} above. Now, the Martin kernel for AR models can be computed as

$$\begin{aligned} k_M(\mathbf{M}_1, \mathbf{M}_2) &= \prod_{i=1}^{n} \cos^2 \theta_i = \det(P_{11}^{-1} P_{12} P_{22}^{-1} P_{21}) \\ &= \frac{|\det(P_{12}^2)|}{|\det(P_{11})||\det(P_{22})|} = \frac{k_d(\mathbf{M}_1, \mathbf{M}_2)}{\sqrt{k_d(\mathbf{M}_1, \mathbf{M}_1)}\sqrt{k_d(\mathbf{M}_2, \mathbf{M}_2)}} = k'_d(\mathbf{M}_1, \mathbf{M}_2). \end{aligned} \tag{26}$$

The last expression is the normalized Binet-Cauchy determinant kernel in eq. (24). ■

Theorem 5.3: For Auto Regressive (AR) models, distances based on the subspace angles can be directly derived from successive maximization of the Binet-Cauchy trace kernels.

Proof: Ignoring the effect of input noise on the Binet Cauchy trace kernel with $\lambda = 1$, consider the optimization problem,

$$\begin{aligned} k'_{tr}(\mathbf{M}_1, \mathbf{M}_2) &= \max_{\mathbf{x}_1, \mathbf{x}_2} \left(\mathbf{x}_1^\top P_{12} \mathbf{x}_2 \right) \\ &\text{subject to} \quad \mathbf{x}_1^\top P_{11} \mathbf{x}_1 = 1 \text{ and } \mathbf{x}_2^\top P_{22} \mathbf{x}_2 = 1, \end{aligned} \tag{27}$$

where P_{ij} is the solution to the Lyapunov equation, $P_{ij} = A_i^\top P_{ij} A_j + C_i^\top C_j$. The Lagrangian of the above optimization problem is,

$$\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2, \mu_1, \mu_2) = \mathbf{x}_1^\top P_{12} \mathbf{x}_2 + \frac{1}{2}\mu_1(1 - \mathbf{x}_1^\top P_{11} \mathbf{x}_1) + \frac{1}{2}\mu_2(1 - \mathbf{x}_2^\top P_{22} \mathbf{x}_2). \tag{28}$$

Differentiating and equating to zero gives,

$$P_{12}\mathbf{x}_2 = \mu_1 P_{11}\mathbf{x}_1, \tag{29}$$

$$P_{12}^\top \mathbf{x}_1 = \mu_2 P_{22}\mathbf{x}_2. \tag{30}$$

Multiplying eq. (29) by $\mathbf{x}_1^\top$ and eq. (30) by $\mathbf{x}_2^\top$ on the right and equating them gives

$$\mu_1 \mathbf{x}_1^\top P_{11}\mathbf{x}_1 = \mathbf{x}_1^\top P_{12}\mathbf{x}_2 = \mathbf{x}_2^\top P_{12}^\top \mathbf{x}_1 = \mu_2 \mathbf{x}_2^\top P_{22}\mathbf{x}_2. \tag{31}$$

Using the constraints in eq. (27), we get $\mu_1 = \mu_2 = \mu$ and thus $\mathbf{x} = [\mathbf{x}_1^\top, \mathbf{x}_2^\top]^\top$ is the solution to the generalized eigenvalue problem:

$$\begin{bmatrix} 0 & P_{12} \\ P_{21} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \mu \begin{bmatrix} P_{11} & 0 \\ 0 & P_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}. \tag{32}$$

Following the construction in [17], μ^2 is the eigenvalue of the matrix $P_{11}^{-1}P_{12}P_{22}^{-1}P_{21}$ and $\mathbf{x} = [\mathbf{x}_1^\top, \mathbf{x}_2^\top]^\top$ is the generalized eigenvector corresponding to the generalized eigenvalue, μ, in eq. (32). Multiplying by $\mathbf{x}^\top$ on both sides of eq. (32), we obtain $\mathbf{x}_1^\top P_{12}\mathbf{x}_2 = \mu$ and thus the solution to the optimization problem becomes

$$k'_{tr}(\mathbf{M}_1, \mathbf{M}_2) = \mu_{\max} = \cos\theta_{\min}, \tag{33}$$

which coincides with the cosine of the smallest subspace angle between systems $\mathbf{M}_1$ and $\mathbf{M}_2$. Analogously, the other subspace angles can be directly computed from the Binet-Cauchy trace kernels by successively solving a constrained optimization problem. In particular, for the k-th smallest subspace angle,

$$\cos\theta_k = \max_{\mathbf{x}_1, \mathbf{x}_2} \left(\mathbf{x}_1^\top P_{12}\mathbf{x}_2\right), \text{ for } k = 2, \ldots, n$$

$$\text{subject to } \mathbf{x}_1^\top P_{11}\mathbf{x}_1 = 1, \mathbf{x}_2^\top P_{22}\mathbf{x}_2 = 1, \tag{34}$$

$$\mathbf{x}_{1;i}^\top P_{11}\mathbf{x}_1 = 0, \mathbf{x}_{2;i}^\top P_{22}\mathbf{x}_2 = 0, \text{ for } i = 1, 2, \ldots, k-1. \tag{35}$$

where $\mathbf{x}_{1;i}$ and $\mathbf{x}_{2;i}$ are the corresponding optimizers for $\cos\theta_i, i = 1, 2, \ldots, k-1$. Hence, the subspace angles between AR systems can be directly derived from the Binet Cauchy kernels with $\lambda = 1$ and therefore, the subspace-angle based distances are special cases of the Binet Cauchy kernels. ■

C. Hybrid metrics on the output means and system dynamics

Another important consideration that is often overlooked is how to incorporate the effect of the temporal means when computing the distances. It is clear that the temporal means of two sequences provide good discriminative power for recognition purposes. Using the temporal means alone as weak classifiers with Boosting has been shown to perform well in [19]. Two simple metrics that can be defined using only the temporal means of the output sequences

$$\text{Norm of the difference of temporal means:} \quad d_p = \|\mu_1 - \mu_2\|^p, \tag{36}$$

$$\text{Polynomial kernel on temporal means:} \quad k_p = |\mu_1^\top \mu_2|^p, \tag{37}$$

where $p \geq 1$ is a free parameter, usually equal to 1.

The distances based on subspace angles, Binet Cauchy kernels and the KL divergence can be combined with the metrics on the temporal means to construct a new class of *hybrid* distance metrics that also give a certain weight to the temporal means when performing recognition. This class of hybrid distances can in general be represented by:

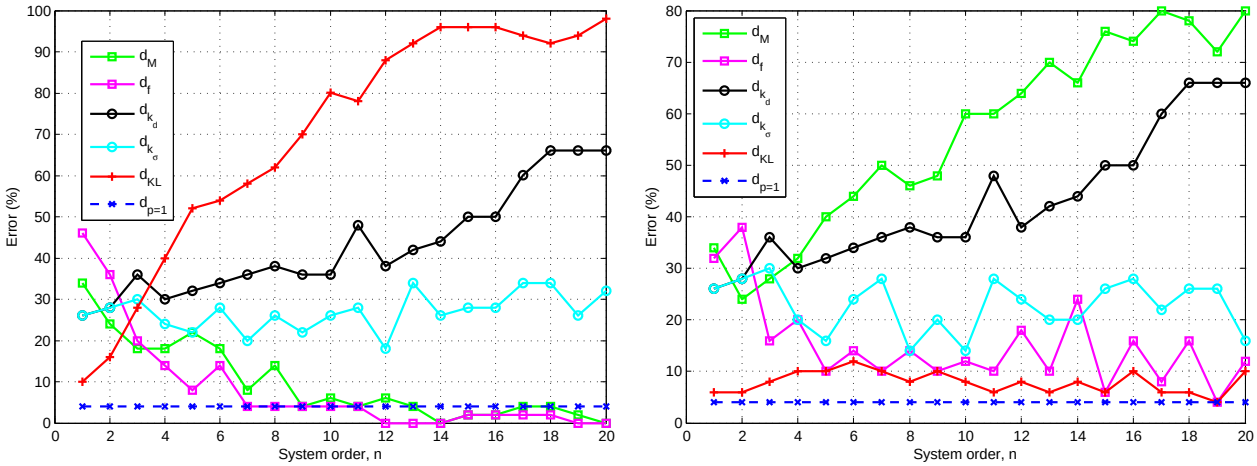
$$d_h(\mathbf{M}_1, \mathbf{M}_2) = (1 - \beta) d_c(\mathbf{M}_1, \mathbf{M}_2) + \beta d_p(\mathbf{M}_1, \mathbf{M}_2), \tag{38}$$

where d_c is any metric between the LDSs and d_p is the distance between the temporal means. Note that d_c and d_p are normalized and scaled such that the maximum distance between any two models in the training set is one. The parameter β is the relative weight between d_c and d_p and can be tuned using cross-validation. Also, notice that for all the metrics in §III, the distances can easily be converted into Radial Basis Function (RBF) kernels with a parameter γ as $k(\mathbf{M}_1, \mathbf{M}_2) = \mathbf{e}^{-\gamma d(\mathbf{M}_1, \mathbf{M}_2)^2}$. This conversion allows γ to be tuned to the specific application using cross-validation during the training phase.

VI. Experimental Evaluation - Standard databases

We now present a benchmark experimental evaluation of the recognition pipeline on three real databases: UCLA dynamic texture database [5], CMU MOCAP human gait database [33] and the MVGL-KOC lip articulation database [16]. These databases have been used extensively in literature, and are a baseline for comparison of any newly developed methods. This section provides an evaluation of the identification methods, metrics and classification methods on these databases.

A. Dynamic texture recognition



(a) SVM-RBF kernels, PCA-ID, no means

(b) SVM-RBF kernels, EM-ID, no means

(c) SVM-RBF kernels, PCA-ID, with means

(d) 1-NN, PCA-ID, with means

Fig. 1. Recognition error for the UCLA dynamic texture database against system order

The UCLA dynamic texture database [5] consists of 200 sequences of various dynamic texture categories including boiling water, fire, flowers, plants, sea and smoke among others. The dataset consists of 50 classes having 4 video sequences each. The videos have 160×110 sized frames with 75 frames per sequence. As in [5], the size of the sequences is reduced to a 48×48 window that captures the key dynamics of the scene. The resulting video sequences for each class are very similar and there is little variation in the appearance from one to the other. As we will show, a naïve metric based only on the temporal means would also perform very well in most cases, however in some cases, using the dynamics would yield even better results. That said, the UCLA dynamic texture database has been the most extensively used database

in dynamic texture literature and thus we also provide benchmark results for this database.

Figure 1 shows the variation of the error percentages across the system order for various metrics for both 1-NN and SVM classification methods. From Figures 1(a) and 1(b), we notice that for both PCA and EM identification methods, the baseline naïve difference of temporal means metric performs very well when used with SVM. However, the Frobenius distance does better than the naïve method for higher system orders. Furthermore, when using PCA as the identification method with the Martin distance and EM with the KL divergence distance at higher orders, we get better than the baseline recognition rates. Also, the Martin distance performs very well when used with PCA identification and poorly when used with EM, whereas KL divergence performs very well with EM and poorly with PCA. The effect of using hybrid metrics can be gauged from Figure 1(c), where the error percentages for all the metrics has decreased greatly. From Figure 1(d), we notice that except for the KL divergence and the maximum singular value kernel, all the other hybrid metrics perform better than the baseline for higher system orders when 1-NN is used.

Table I(a) provides the percentage of recognition error for the UCLA dynamic textures database when leave-one-out 1-NN is used as the classification method. Table I(b) provides the mean and standard deviation of recognition errors for 10 randomized trials when SVM was used as the classification method, as in the case of human gait experiments above. For SVM, 3 out of the 4 (75%) sequences were used for training and the rest for testing. The system order used in both cases is $n = 9$. Clearly, using the temporal means improves the performance of all the metrics in the case of 1-NN. However, only the hybrid Frobenius distance performs better than the baseline metric. The hybrid Martin distance and the determinant kernel also perform as well as the naïve metric when used with PCA. From table I(b), we can see that except for a few cases (the regular determinant kernel and Martin distance with EM), the recognition errors have decreased for all the metrics, when compared to their 1-NN counterparts in table I(a), as a result of using SVM which is a more sophisticated classification method. The best metric/identification methods are the hybrid Frobenius distance with both EM and PCA identification, the Martin distance with PCA identification and the maximum singular value kernel with EM identification. All of these have means and standard deviations of errors equal to or less than the naïve temporal means metric and hence the metrics/identification methods of choice.

TABLE I

MEAN AND STANDARD DEVIATION OF ERROR PERCENTAGES FOR DYNAMIC TEXTURE RECOGNITION

(a) $n = 9$, 1-NN, leave one out classification

	Without mean		With mean	
ID	PCA	EM	PCA	EM
d_M	18	51	7	35
d_f	19	23	6	6
d_{k_d}	15	15	7	8
d_{k_σ}	42	36	11	10
d_{KL}	78	39	12	22
$d_{p=1}$			7	7

(b) $n = 9$, SVM, 75% training, 25% testing

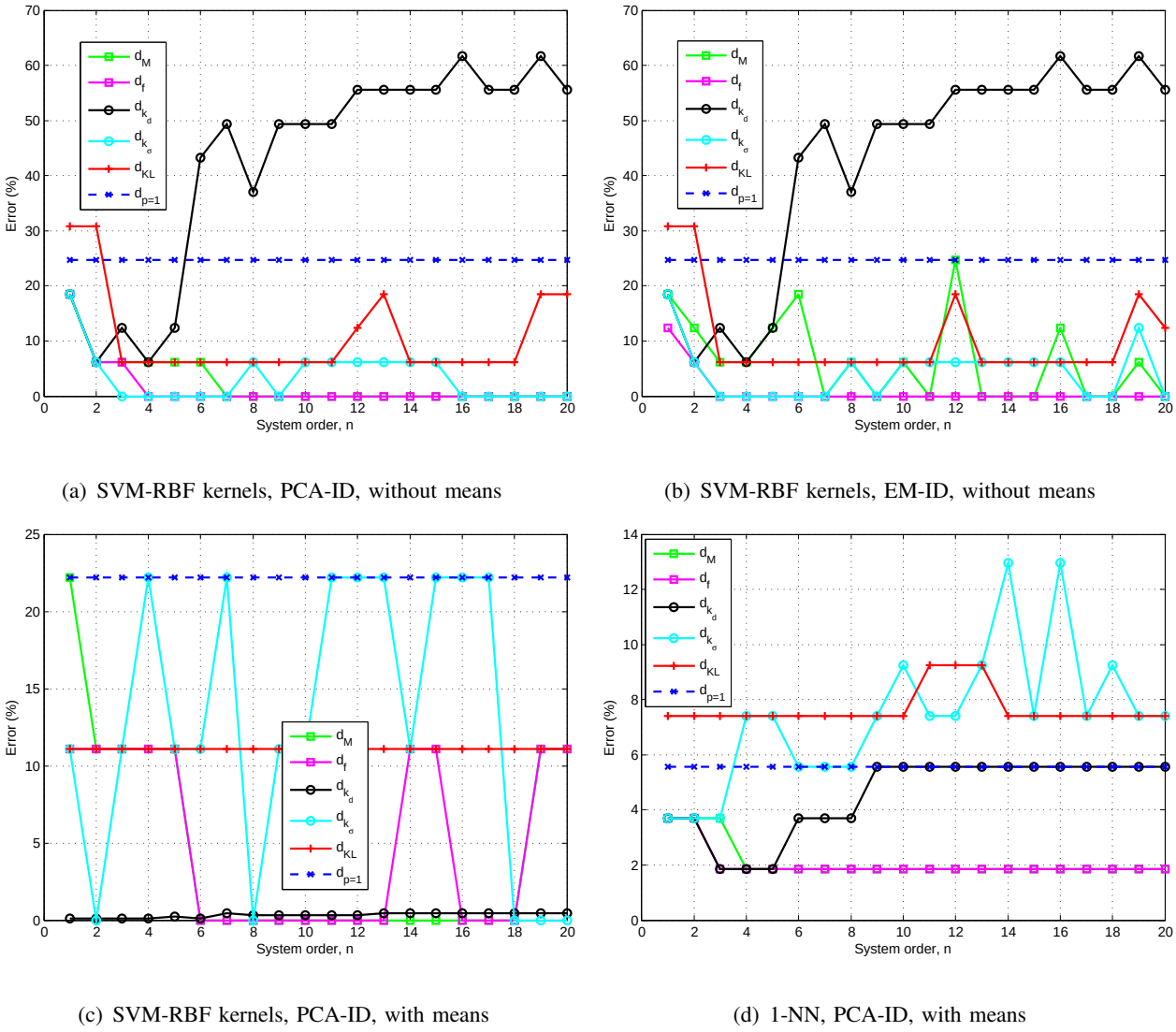
	Without mean		With mean	
ID	PCA	EM	PCA	EM
d_M	6 ± 3	52 ± 2	3 ± 1	19 ± 6
d_f	5 ± 3	12 ± 5	3 ± 2	4 ± 2
d_{k_d}	27 ± 5	29 ± 7	6 ± 1	6 ± 2
d_{k_σ}	25 ± 4	18 ± 3	5 ± 4	2 ± 2
d_{KL}	67 ± 4	6 ± 4	8 ± 2	6 ± 1
$d_{p=1}$			4 ± 2	4 ± 2

B. Human gait recognition

The CMU MOtion CAPture (MOCAP) database [33] consists of motion capture data from a number of human actions characterized as locomotion, physical activities, interaction with the environment as well as with people. We consider a subset of the human locomotion data that consists of a total of 54 sequences with 18 sequences each of the three classes: running, walking and jumping. Each sequence consists of 52 joint angle trajectories and 6 degrees of freedom for a skeletal model, recorded using the motion of markers placed on the body of each subject.

Figure 2 shows the recognition error percentage against system order for the CMU MOCAP database. Figures, 2(a) to 2(c) show the results when SVM is used as the classification method. The first 15 of the 18 (approx 83%) total sequences per class were chosen as the training set and the last 3 were chosen as the testing set. Irrespective of the identification method, all the metrics, except the determinant kernel, consistently give better recognition results than the base line difference of temporal means metric. Also, all the hybrid metrics always perform better than the regular ones. Overall the best metrics are the Martin distance, the Frobenius distance and the maximum singular value kernel which give 0% recognition error for several system orders. As Figure 2(d) shows, when using 1-NN with means, only the Martin and Frobenius distances stay below the baseline and consistently perform below 2% recognition error.

Table II(a) provides the mean error percentages for leave-one-out 1-NN classification. Recognition



(a) SVM-RBF kernels, PCA-ID, without means

(b) SVM-RBF kernels, EM-ID, without means

(c) SVM-RBF kernels, PCA-ID, with means

(d) 1-NN, PCA-ID, with means

Fig. 2. Recognition error for the CMU MOCAP database against system order

error of 0% is achieved with the original Martin (without adding temporal means) distance and the Binet Cauchy determinant kernel for both PCA and EM identification methods. Both the original and the hybrid Frobenius distance and the hybrid maximum singular value kernel perform better than using the temporal mean alone with PCA and EM identification.

Table II(b) provides the mean and standard deviation of error percentages for 10 runs of SVM classification where the training and testing sequences were randomly chosen. Both these results show the consistency of a metric/identification method, when used for human gait recognition. We get 0% mean recognition error with 0% standard deviation when using PCA with the original Martin and Frobenius distances. The Frobenius distance also gives 0% error when used with EM. The performance in general decreases, except for the maximum singular value kernel, when using the hybrid metrics. However the hybrid Martin and

Frobenius distances as well as the hybrid maximum singular kernel, still perform better than using the naïve temporal means metric when used with PCA or EM identification. This shows that considering the dynamics is important in recognizing human gaits and using a naïve metric based only on temporal means of the trajectories will not always perform the best.

TABLE II

MEAN AND STANDARD DEVIATION OF ERROR PERCENTAGES FOR HUMAN GAIT RECOGNITION

(a) CMU-MOCAP - $n = 7$, 1-NN, leave one out classification

	Without mean			With mean		
ID	N4SID	PCA	EM	N4SID	PCA	EM
d_M	18	0	0	6	2	4
d_f	56	2	2	6	2	2
d_{k_d}	7	0	0	6	4	4
d_{k_σ}	14	11	9	4	6	4
d_{KL}	26	7	7	6	7	9
$d_{p=1}$				6	6	6

(b) CMU-MOCAP - $n = 7$, SVM, 83% training, 17% testing

	Without mean			With mean		
ID	N4SID	PCA	EM	N4SID	PCA	EM
d_M	20 ± 10	0 ± 0	4 ± 14	7 ± 8	4 ± 6	4 ± 7
d_f	17 ± 8	0 ± 0	0 ± 0	7 ± 8	4 ± 6	2 ± 5
d_{k_d}	27 ± 16	20 ± 9	27 ± 9	10 ± 11	9 ± 14	13 ± 9
d_{k_σ}	10 ± 8	11 ± 7	17 ± 13	4 ± 6	5 ± 8	3 ± 5
d_{KL}	60 ± 18	9 ± 15	4 ± 6	63 ± 9	24 ± 18	7 ± 6
$d_{p=1}$				7 ± 6	7 ± 6	7 ± 6

C. Lip articulation recognition

The MVGL-KOC lip articulation database [16] consists of videos of a number of subjects uttering their names and digits as passwords. We consider a subset of the database that contains 80 sequences with 10 sequences each of 8 different speakers uttering their names. The position of 32 landmarks around the upper and lower lip is extracted from the frontal view of the subjects and tracked over time through the video. These landmarks are considered as the output of a LDS and the goal of the recognition algorithm is to identify the speaker from a novel video. This is a particularly difficult database to perform recognition on, as shown by the results in [16]. Due to inherent non-linearities in landmark trajectories on the lips and thus, errors around 10-20% are considered to be very good.

Tables III(a) and III(b) display mean errors and standard deviations for 1-NN and SVM classification respectively. We notice that when using 1-NN, the regular Frobenius distance used with EM gives the best recognition result of 21%. The Frobenius distance and the KL divergence distance with PCA and the hybrid Frobenius distance with EM have similar error percentages in the low 20%s. When using

SVM as the classification method, the regular Frobenius distance when used with EM performs the best and gives the lowest mean recognition error of 14% over 10 trials with randomly chosen training and test sets of size 7 and 3 sequences per class respectively. Both the hybrid Frobenius distance and the KL divergence distance when used with PCA perform very well. Here, we particularly observe the case that a naïve temporal means metric does not work at all. The lowest error percentages achieved when using the temporal means metric with SVM is 67% which is significantly larger than the lowest error percentage achieved with the Frobenius distance and the KL divergence distance. Hence, dynamics of the lip articulation process are extremely important when designing a recognition framework.

TABLE III

MEAN AND STANDARD DEVIATION OF ERROR PERCENTAGES FOR LIP ARTICULATION RECOGNITION

(a) $n = 3$, 1-NN, leave one out classification

	Without mean			With mean		
ID	N4SID	PCA	EM	N4SID	PCA	EM
d_M	44	41	33	65	63	34
d_f	45	25	21	56	56	24
d_{k_d}	31	34	29	40	41	28
d_{k_σ}	59	53	56	60	61	55
d_{KL}	64	23	26	76	70	35
$d_{p=1}$				80	80	80

(b) $n = 3$, SVM, 70% training, 30% testing

	Without mean			With mean		
ID	N4SID	PCA	EM	N4SID	PCA	EM
d_M	43 ± 5	40 ± 8	46 ± 14	58 ± 6	47 ± 8	49 ± 9
d_f	19 ± 6	17 ± 6	14 ± 6	32 ± 10	15 ± 7	28 ± 6
d_{k_d}	34 ± 11	33 ± 11	29 ± 6	35 ± 8	28 ± 6	38 ± 7
d_{k_σ}	51 ± 11	38 ± 7	50 ± 10	53 ± 8	39 ± 6	49 ± 11
d_{KL}	39 ± 8	15 ± 5	20 ± 12	73 ± 8	28 ± 4	52 ± 4
$d_{p=1}$				67 ± 10	67 ± 10	67 ± 10

VII. EXPERIMENTAL EVALUATION - ROBUSTNESS OF THE RECOGNITION PIPELINE

So far, we have ignored any effects that variations in observation and experimental conditions might have on the performance of the recognition algorithms. Changes in observation conditions include changes in spatial scale, frequency and measurement noise of the observations. Changes in spatial scale, for example, can manifest in videos of dynamic textures. The appearance of water observed from near is very different from that when observed from far. Similarly, changes in the frequency of observation arise due to different frame rates of videos of dynamic textures, human activity or lip movements. Changes in noise-level can be introduced by a tracking algorithm that is faster for real-time performance but that is more error-prone to noise in the device e.g., in the case of human activities or lip articulation or just noise in the video

e.g., in the case of dynamic textures. Changes in experimental conditions, on the other hand, affect the training stages of the classification algorithms. Factors such as the system order used during the system identification stage, the amount of training data available, the number of samples per time-series data used for learning and the number of classes are important when considering the robustness of a set of recognition algorithms. A metric is superior for recognition purposes if the misclassification rate does not vary widely across these changes.

Although the metrics and algorithms introduced in the previous sections are not explicitly designed to deal with these changes, it is important to know which choices are more robust to these variations. The goal of this section is to systematically analyze the performance of the recognition pipeline under changes in the two aforementioned scenarios.

A. Setup

In order to test the robustness of the recognition pipeline to variations in the observation conditions, we look at the performance of the metrics under the following changes:

1) Spatial scale,
2) Temporal scale, i.e., frequency of observation, and
3) Noise-level.

To test the robustness to variations in experimental conditions, we look at the changes in the following scenarios:

1) Length of sequences used for learning the system parameters,
2) Amount of training data used for classification, and
3) Number of classes.

For the purpose of evaluation of the recognition pipeline with respect to each of the above variations, we require a large database that has a labeled set of sequences exhibiting the above variations in observation conditions. To evaluate the robustness to experimental conditions, we also require a large number of classes with a large number of sequences per class. Furthermore, we require a large number of samples per sequence. Unfortunately, such a database does not exist. To overcome this difficulty, we used the DynTex dynamic textures database [34] and generated a large number of synthetic sequences that are

representative of changes in observation and experimental conditions for any general LDS. Some sample frames from the database are shown in Figure 3. The original database consists of a large number of 25 fps 720×576 frame-size videos of various dynamical visual phenomena. To test the robustness of the

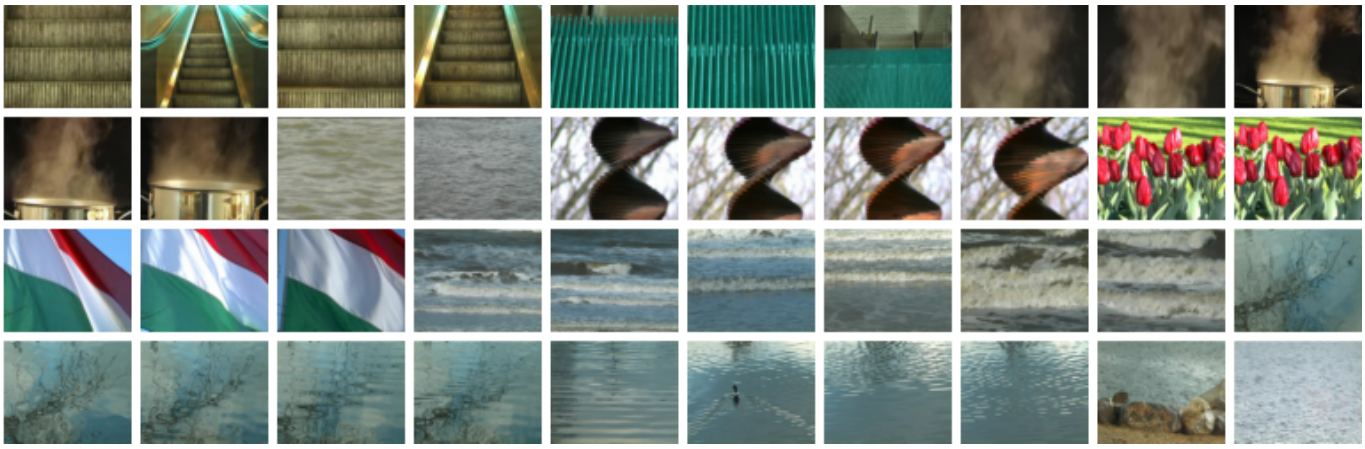
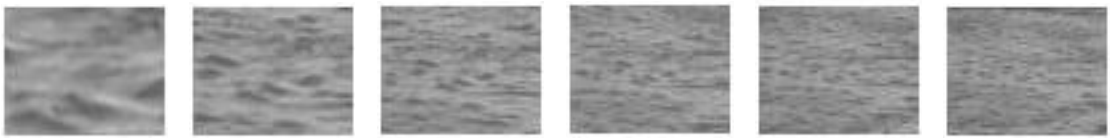


Fig. 3. Sample thumbnails from the DynTex database

recognition pipeline to changes in all three observation conditions and the length of sequences used for learning, we chose a subset of three classes; water, steam and escalator. To test the robustness of the recognition pipeline to changes in the amount of training data and the number of classes, we chose another subset of 14 classes. These are described as follows:

To synthetically generate space-scale variations, we took 16 sequences from each of the three classes – water, steam and escalator – and generated six different scales by taking increasingly larger sized videos of a central portion of the original sequences. We then downsampled these videos to the size of this central portion. More specifically, from a 360×288 pixel sequence cropped from the original 720×576 sequence, we took the central 60×48 portion to represent the original scale-level 1. Scale-levels 2 to 6 similarly are downsampled versions of the central 120×96 to 360×288 portions of this video respectively. Figure 4 shows an illustration for the various scaled versions of the water video. To synthetically generate time-scale, i.e., frequency variations, we resample and linearly interpolate the original video sequences at decreasing rates. The original time-scale is fixed at 30 frames and subsequent levels are generated by downsampling and interpolating 40, 50, 60 and 70 frames to 30 frames. This creates lower frequency observations of the original dynamical process. Finally, to synthetically generate LDSs with different system and observation noise, we resynthesize the sequences from the learnt system parameters of the original sequences by

varying the noise variance of the system noise, $\mathbf{v}$, and the output noise, $\mathbf{w}$, in equation (1). We synthesize sequences with i.i.d. Gaussian noise with standard deviations 1-5 for this purpose.



(a) Original scale (b) Scale-2 (c) Scale-3 (d) Scale-4 (e) Scale-5 (f) Scale-6

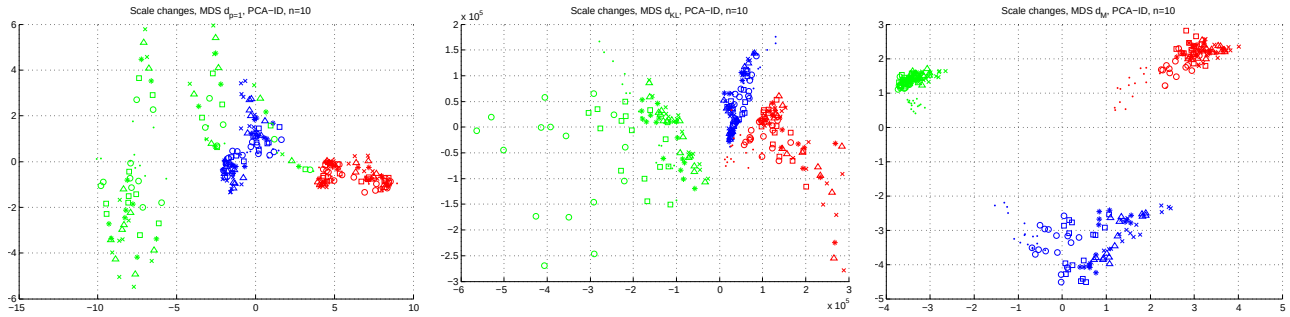
Fig. 4. Space-scaled thumbnails of the water sequence from original scale at the left to largest at the right

To test how the length of sequences used during the learning stage affects the performance of the algorithms, we use sequences of length kn samples, where n is the system order and $k \in \{2, 3, 4, 5, 6, 7\}$. Notice that n is the minimum number of samples required for system identification, however, having exactly n samples often makes the matrix factorization step in the system identification procedure numerically unstable and hence we do not use $k = 1$ for this set of experiments. To evaluate the performance of the recognition pipeline against changes in the amount of training data and the number of classes, we choose another set of 14 different classes and re-sample and crop the original sequence to form 50 sequences for each class. This provides us with a large dataset of a total of 700 sequences of size 144×180 pixels with 75 frames each. To compare the the performance w.r.t. the variation in the number of training samples, we divide the sequences for each class into 5 subsets of 10 sequences each. Training is performed using the first 10, 20, 30 and 40 sequences respectively to represent the variation in the amount of training data whereas the testing is performed on the last 10 sequences in each case. To determine the performance of all the metrics against the number of classes, we test the algorithms with the number of classes ranging from 2 to 14.

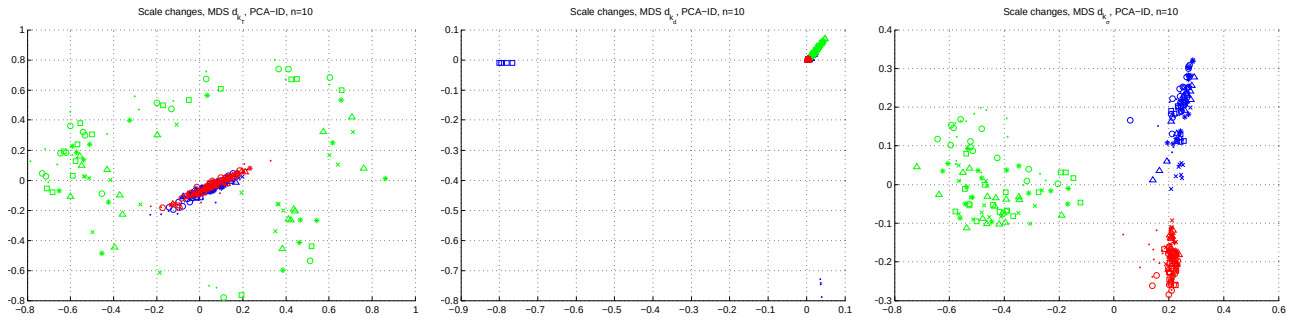
Although we use sequences from the DynTex database to generate synthetic sequences for our evaluation, this treatment is very generic and is applicable to any LDS. Also, to the best of our knowledge, such a detailed robustness evaluation has never been performed in the LDS-based classification and recognition literature.

B. Results

1) Changes in Observation Conditions: Given pairwise distances between a number of models, we can use classical Multi-Dimensional Scaling (MDS) [35], to get a low-dimensional representation of these models as points in a Euclidean space, that have the same distances as the original models. A good metric will result in models of the same class lying close to each other in this low-dimensional space whereas models of different classes will lie far away from each other. The best metric would be the one that separates the classes perfectly, i.e., models of the same class would be clustered together and would not overlap with other clusters. Figure 5 shows the results of performing MDS on several distance metrics on the dataset with varying spatial scale. We compute the distance between any two models in the training set. The matrix of distances is then factorized as $D = UU^\top$ using a rank 2 SVD. Each row of U gives a 2-D vector plotted in the figures. We immediately see that the naïve, difference of temporal means



(a) Temporal Means (b) KL-divergence (c) Martin distance



(d) Trace kernel with initial conditions (e) Determinant kernel (f) Maximum singular value kernel

Fig. 5. Classical Multi-dimensional scaling results for different distance metrics on scale change data. Color codes: red = water, green = steam, blue = escalator. Scale markers: dot = original scale, circle = scale 2, square = scale 3, * = scale 4, triangle = scale 5, x = scale 6.

metric, in Figure 5(a) does not provide good separability of the three classes. The KL divergence distance and the original initial-conditions dependent trace kernel proposed in [18], as shown in Figures 5(b) and

5(d) respectively, also do not provide good class separability. However, these metrics are all robust to changes in spatial scale as systems of the same class cluster together independent of their spatial scale. The Martin distance in Figure 5(c) provides perfect class separability and clusters the systems in all three classes independent of their spatial scale. We can see the effects of initial-state independent kernels from Figure 5(e) and 5(f). The determinant kernel in Figure 5(e) performs poorly. The maximum singular value kernel, on the other hand, performs at par with the Martin distance and clusters all three classes correctly independent of changes in spatial scale.

Figures 6(a)-6(f) show the error percentages of recognition for various metrics and classification methods for the 3-class recognition problem with different spatial scales. A test sequence is correctly classified if the correct class-label is assigned during the testing phase, independent of the scale. This demonstrates robustness to changes in spatial scale of the data. For each system order, training is performed using only sequences of the same scale level. Testing is then performed on sequences of all scale-levels. We repeat this over all scale-change levels and the means and standard deviations of the recognition errors are reported. Figures 6(a) and 6(b) show the mean and standard deviation of the recognition errors for several hybrid metrics (temporal means included in metric computation). We notice that the KL divergence distance, the Martin distance and the maximum singular value kernel give the best recognition performance consistently over all system orders. We conclude that these hybrid metrics are robust to changes in spatial scale when used with SVMs as the classification method. Figure 6(c) shows the mean error percentages when the temporal means are not included in the metric computation. The error-percentages for all but the KL-divergence and Martin distances go up drastically. Thus using the temporal mean in the metric calculation is crucial for the Binet-Cauchy kernels. Figure 6(d) shows that using EM as the system identification method compared to PCA gives somewhat worse results, however the metrics of choice are still the same as in Figure 6(a). Figure 6(e) demonstrates that 1-NN classification performs worse than SVM classification. Finally, Figure 6(f) shows a specific instance when training is performed using the sequences at the original scale-level and tested against increasing scales when using hybrid metrics with SVM and PCA as the system identification method for a system order of $n = 10$. We again see that KL-divergence, Martin distance and the maximum singular value kernel consistently give 0% error and hence should be the metrics of choice when observations from the dynamic visual phenomenon vary in

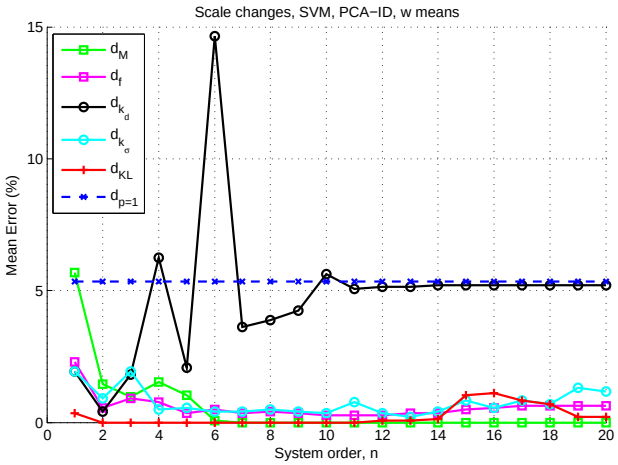
spatial-scale.

Figure 7(a) shows the mean error percentages across system order for various hybrid metrics when used for recognition in the 3-class recognition problem with sequences containing different frequencies of observation. The training and testing was performed analogously as for the spatial-scale variation experiments just described. We again see that the maximum singular value kernel, the KL-divergence distance and the Martin distance perform the best. In fact, using the Martin distance, we get 0% error across all orders. We can again see that all the metrics, except the determinant kernel, perform much better than the naïve difference of temporal means metric. Figure 7(b) shows the error percentages when training is performed on the original sequence and testing is done on sequences with decreasing frequencies. Again, the KL-divergence distance, Martin distance and the maximum singular value kernel are the metrics of choice.

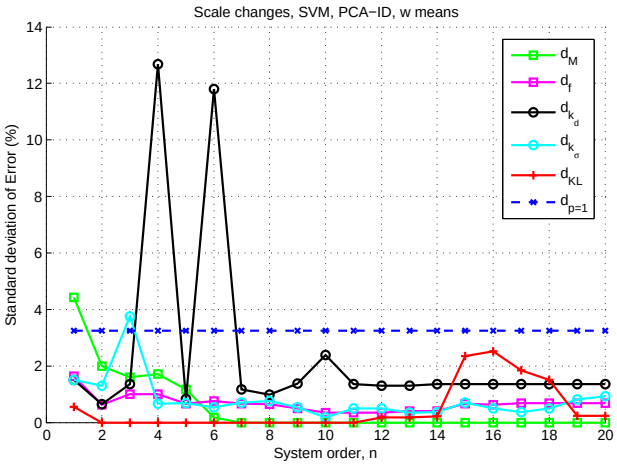
Figure 8 shows the recognition results for the 3-classes above, when the sequences are observed at various noise-levels. We again see that the Martin distance, maximum singular value kernel and the KL divergence kernel perform the best across system order. Figure 8(b) shows an instance of the experiment when training is performed with the original sequences and testing is performed over the same sequences with different noise-levels. Clearly the best metrics are invariant to changes in noise-level of the sequences.

From the above experiments, it is clear that the LDS recognition pipeline is robust to changes in spatial-scale, frequency of observation and noise-level in the observed sequences, when used with PCA identification, the Martin distance, the maximum singular value kernel or the KL divergence distance, and SVM classification.

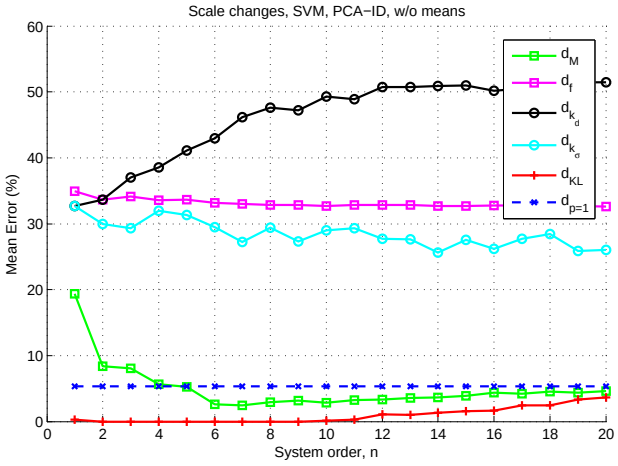
2) Changes in Experimental Conditions: Figure 9 provides the mean error percentages for various metrics when the system parameters are learnt from sequences of different lengths. The length of the sequence from which the system parameters of a LDS are learnt is an important consideration. From Figure 9(a), we notice that as the system order increases, the mean error percentage in general decreases for all metrics. On the other hand, Figure 9(b) shows that when training is performed with sequences of length $2n$, where n is the system order, only the Martin distance gives 0% recognition error when tested against systems with parameters learnt from a larger sequence size. The KL divergence distance requires the length of the sequences to be the same across systems and hence was not tested in this experiment.



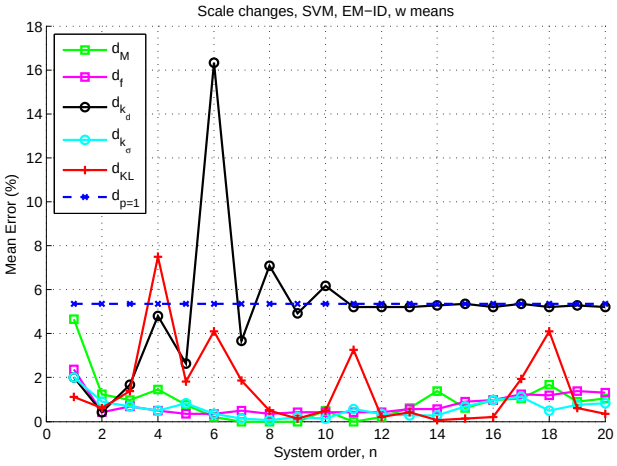
(a)



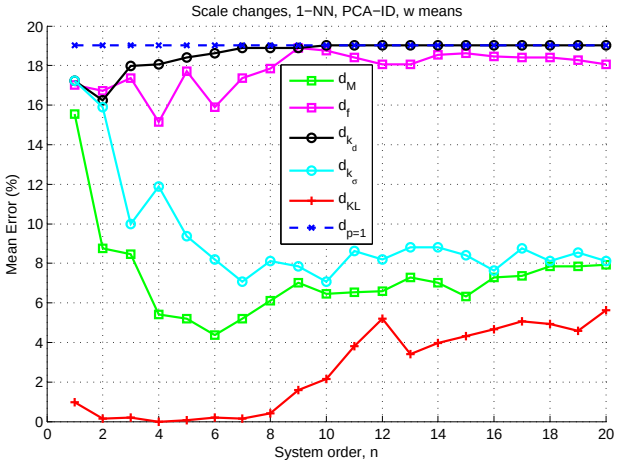
(b)



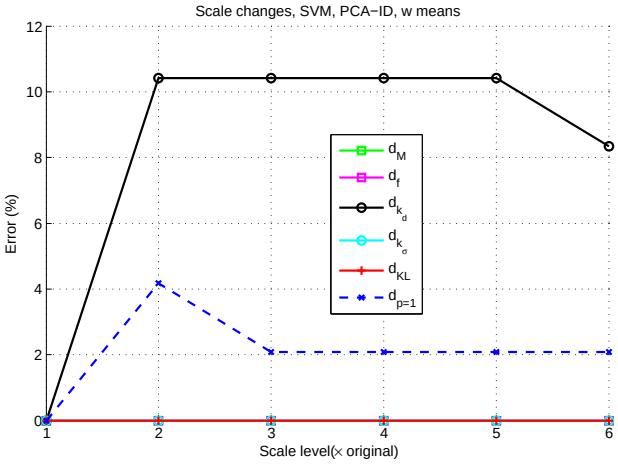
(c)



(d)

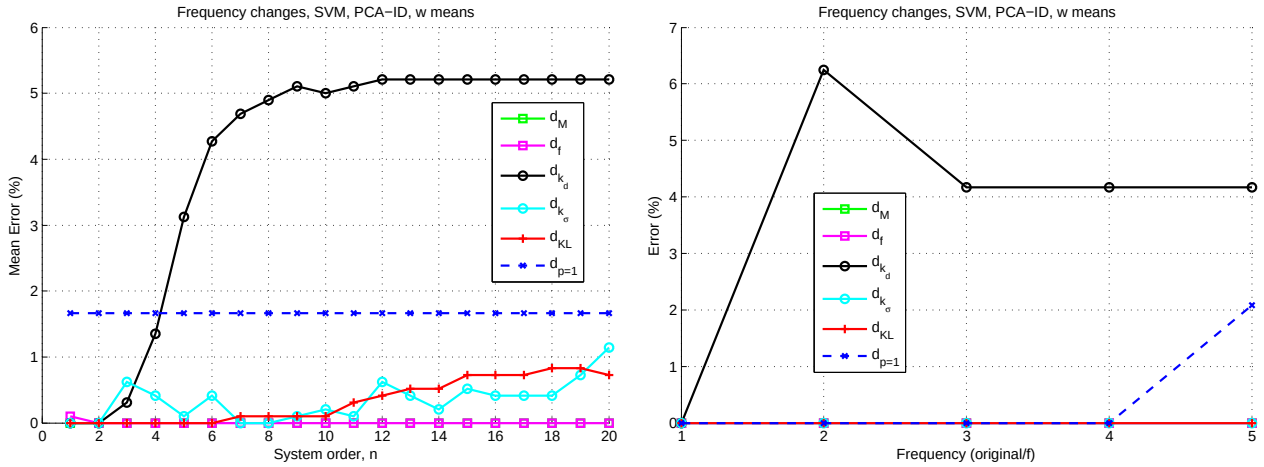


(e)



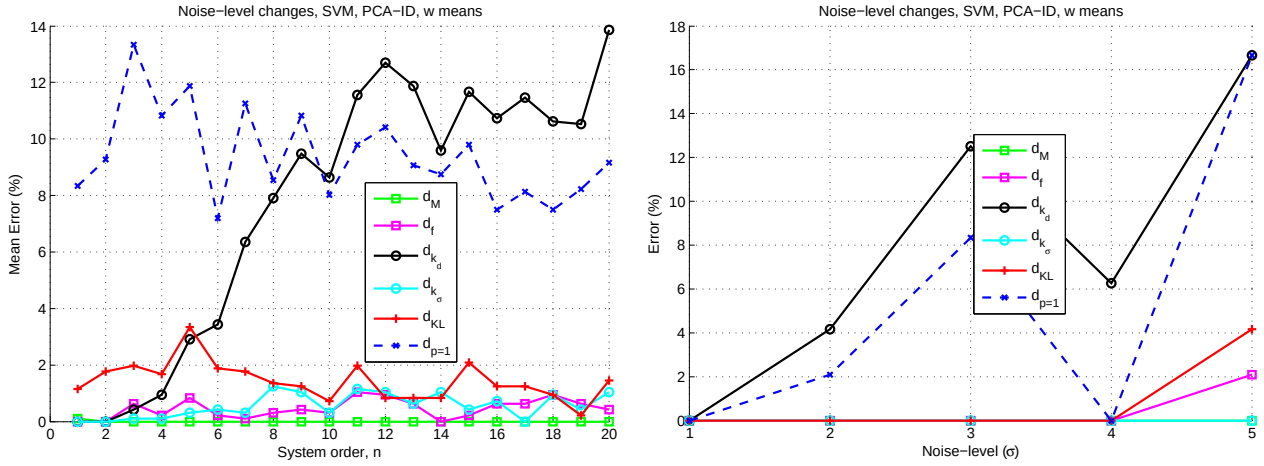
(f)

Fig. 6. Performance of metrics against changes in spatial scale



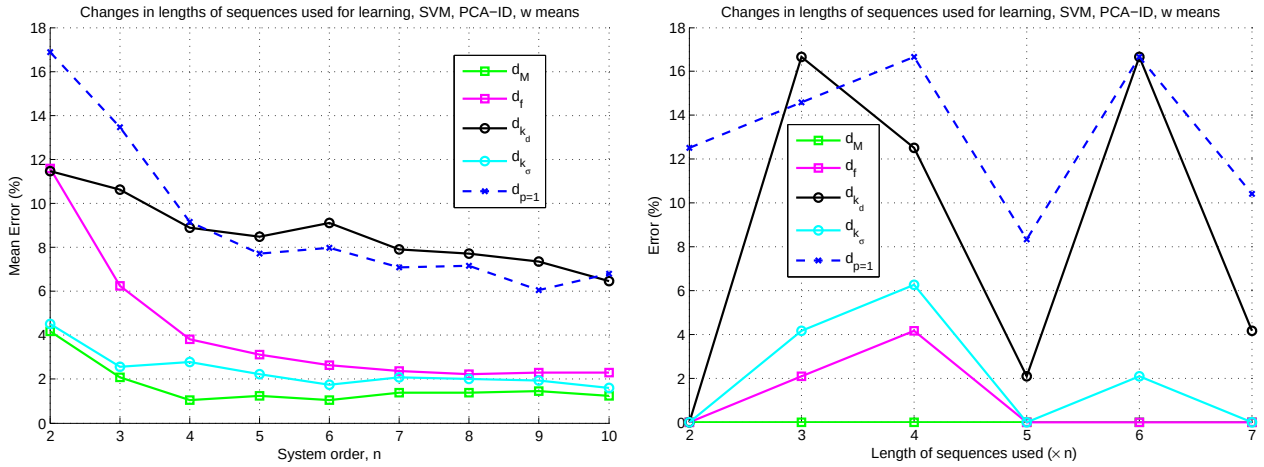
(a) (b)

Fig. 7. Performance of metrics against changes in Frequency



(a) (b)

Fig. 8. Performance of metrics against changes in noise-level

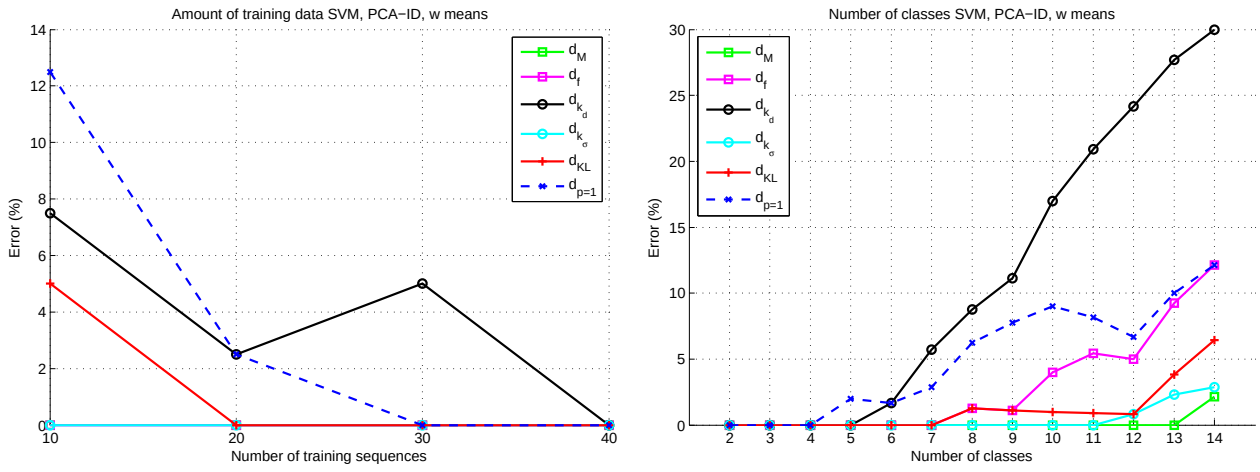


(a) (b)

Fig. 9. Performance of metrics against changes in the length of sequences used for learning

Figure 10(a) provides error percentages for a 3-class problem when the amount of training data is varied. Out of the 50 sequences per class, training is performed on 10, 20, 30 and 40 sequences and for each case, testing is performed on the last 10 sequences. The mean and standard deviation of errors of these experiments is reported. We observe that the amount of training affects the recognition rate: the more training data is available, the better the recognition results. The Martin distance consistently give 0% errors for all training data sizes.

Until now, all the experiments in this section were based on recognition of three classes. To investigate how the number of classes affects the recognition pipeline, Figure 10(b) provides error percentages when the number of classes is varied from 2 to 14. Training is performed on 40 sequences from each class and testing is performed on the last 10 sequences. We can see that the recognition performance of all the metrics becomes poorer as the number of classes increases. The Martin distance, the maximum singular value kernel and the KL divergence distance still prove to be the most resilient to changes in the number of classes whereas the the naïve temporal means method does not show this property.



(a) (b)

Fig. 10. Performance of metrics against changes in the amount of training data and the number of classes

From the above experiments we can see that one needs to consider both the effects of changes in observation and experimental conditions on the recognition pipeline. Any given metric might not be suitable in all scenarios. In particular, we showed that the error percentages of recognition results from each of the three classes of metrics proposed in sections §III and §V. We observed that recognition rates were affected by changes in observation and experimental conditions however the Martin distance, the

Binet Cauchy maximum singular value kernel and the KL-divergence distance seemed to be the most robust out of all the other metrics.

VIII. Conclusions

We have presented an exhaustive study and experimental evaluation of LDS-based recognition algorithms. We have derived three new kernels based on the Binet-Cauchy kernels that are independent of the initial conditions of the models. We have also shown that that kernels based on subspace angles are particular cases of the Binet Cauchy kernels. We have studied in detail the effect of the temporal means of the output sequences on the performance of algorithms and found that using the means is critical for good performance in human gaits and dynamic texture recognition. However the naïve, difference of temporal means, is not a good metric and the dynamics of the system have to be taken into account to get the best recognition results. We have considered the effects of several observation and experimental variations on the performance of the recognition pipeline and empirically found that certain metrics from all the three classes inherently are robust to these variations. The empirical results provided in the paper suggest that the Martin and Frobenius distance, the maximum singular value kernel and the KL divergence distance prove to be the best and most robust metrics in most scenarios. We hope that the results in this paper will be extremely valuable towards future algorithm development, testing and benchmarking.

Acknowledgment

This work was partially supported by startup funds from JHU, by grants ONR N00014-05-10836, NSF CAREER 0447739, NSF EHS-0509101, and by contract JHU APL-934652.

References

[1] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka, "Visual categorization with bags of keypoints," in *European Conference on Computer Vision*, 2004.

[2] E. B. Sudderth, A. B. Torralba, W. T. Freeman, and A. S. Willsky, "Learning hierarchical models of scenes, objects, and parts," in *International Conference on Computer Vision*, 2005, pp. 1331–1338.

[3] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[4] X. Lan and D. P. Huttenlocher, "Beyond trees: Common-factor models for 2d human pose recovery," in *IEEE International Conference on Computer Vision*, 2005.

[5] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, "Dynamic textures," *Int. Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[6] G. Doretto and S. Soatto, "Editable dynamic textures," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. II, 2003, pp. 137–142.

[7] L. Yuan, F. Wen, C. Liu, and H. Shum, "Synthesizing dynamic texture with closed-loop linear dynamic system," in *European Conf. on Computer Vision*, 2004, pp. 603–616.

[8] P. Saisan, A. Bissacco, A. Chiuso, and S. Soatto, "Modeling and synthesis of facial motion driven by speech," in *European Conf. on Computer Vision*, vol. 3, 2004, pp. 456–467.

[9] A. Chan and N. Vasconcelos, "Probabilistic kernels for the classification of auto-regressive visual processes," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 846–851.

[10] G. Doretto, D. Cremers, P. Favaro, and S. Soatto, "Dynamic texture segmentation," in *IEEE Int. Conf. on Computer Vision*, 2003, pp. 44–49.

[11] A. Ghoreyshi and R. Vidal, "Segmenting dynamic textures with Ising descriptors, ARX models and level sets," in *International Workshop on Dynamic Vision*, ser. LNCS 4358, 2006, pp. 127–141.

[12] C. Bregler, "Learning and recognizing human dynamics in video sequences," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 568–574.

[13] A. Bissacco, A. Chiuso, Y. Ma, and S. Soatto, "Recognition of human gaits," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2001, pp. 52–58.

[14] C. Cohen, L. Conway, and D. Koditschek, "Dynamical system representation, generation, and recognition of basic oscillatory motion gestures," in *International Conference on Automatic Face and Gesture Recognition*, 1996, pp. 60 – 65.

[15] G. Aggarwal, A. Roy-Chowdhury, and R. Chellappa, "A system identification approach for video-based face recognition," in *IEEE Int. Conf. on Pattern Recognition*, 2004, pp. 23–26.

[16] H. Çetingül, R. Chaudhry, and R. Vidal, "A system theoretic approach to synthesis and classification of lip articulation," in *International Workshop on Dynamical Vision*, 2007.

[17] K. D. Cock and B. D. Moor, "Subspace angles and distances between ARMA models," *System and Control Letters*, vol. 46, no. 4, pp. 265–270, 2002.

[18] S. Vishwanathan, A. Smola, and R. Vidal, "Binet-Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes," *International Journal of Computer Vision*, vol. 73, no. 1, pp. 95–119, 2007.

[19] R. Vidal and P. Favaro, "Dynamicboost: Boosting time series generated by dynamical systems," in *IEEE International Conference on Computer Vision*, 2007.

[20] F. Woolfe and A. Fitzgibbon, "Shift-invariant dynamic texture recognition," in *European Conference on Computer Vision*, 2006, pp. II: 549–562.

[21] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto, "Dynamic texture recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. II, 2001, pp. 58–63.

[22] R. Shumway and D. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *Journal of Time Series Analysis*, vol. 3, no. 4, pp. 253–264, 1982.

[23] P. V. Overschee and B. D. Moor, "Subspace algorithms for the stochastic identification problem," *Automatica*, vol. 29, no. 3, pp. 649–660, 1993.

[24] P. van Overschee and B. D. Moor, *Subspace Identification for Linear Systems*. Kluwer Academic Publishers, 1996.

[25] D. Bauer, "Asymptotic properties of subspace estimators," *Automatica*, vol. 41, no. 3, pp. 359–376, 2005.

[26] A. Bissacco, A. Chiuso, and S. Soatto, "Classification and recognition of dynamical models: The role of phase, independent components, kernels and optimal transport," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 1958–1972, 2006.

[27] T. Kailath, *Linear Systems*. Prentice Hall, 1980.

[28] H. Golub and C. V. Loan, *Matrix Computations*, 2nd ed. Johns Hopkins University Press, 1996.

[29] L. Wolf and A. Shashua, "Learning over sets using kernel principal angles," *Journal of Machine Learning Research*, vol. 4, no. 10, pp. 913–931, 2003.

[30] B. Schölkopf and A. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.

[31] A. Ravichandran and R. Vidal, "Video registration using dynamic textures," in *European Conference on Computer Vision*, 2008.

[32] A. Martin, "A metric for ARMA processes," *IEEE Trans. on Signal Processing*, vol. 48, no. 4, pp. 1164–1170, 2000.

[33] CMU, "Mocap database. http://mpcap.cs.cmu.edu," 2003. [Online]. Available: http://mpcap.cs.cmu.edu

[34] R. Péteri, M. Huskies, and S. Fazekas, "Dyntex: A comprehensive database of dynamic textures," online Dynamic Texture Database. [Online]. Available: www.cwi.nl/projects/dyntex/

[35] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Springer, 2003.